



(11)

EP 1 527 568 B1

(12)

EUROPEAN PATENT SPECIFICATION

(45) Date of publication and mention of the grant of the patent:
19.09.2018 Bulletin 2018/38

(21) Application number: **03735876.9**

(22) Date of filing: **20.06.2003**

(51) Int Cl.:

H04L 12/701 ^(2013.01) **H04L 12/801** ^(2013.01)
H04L 12/863 ^(2013.01) **H04L 12/873** ^(2013.01)
H04L 12/773 ^(2013.01) **H04L 12/857** ^(2013.01)
H04L 12/869 ^(2013.01) **H04L 12/851** ^(2013.01)

(86) International application number:
PCT/IB2003/002426

(87) International publication number:
WO 2004/004248 (08.01.2004 Gazette 2004/02)

(54) **PROGRAMMABLE SCHEDULING FOR IP ROUTERS**

PROGRAMMIERBARER PLAN FÜR IP-ROUTER

ORDONNANCEMENT PROGRAMMABLE POUR ROUTEURS IP

(84) Designated Contracting States:
AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HU IE IT LI LU MC NL PT RO SE SI SK TR

(30) Priority: **26.06.2002 US 183057**

(43) Date of publication of application:
04.05.2005 Bulletin 2005/18

(73) Proprietor: **Nokia Technologies Oy**
02610 Espoo (FI)

(72) Inventors:
• **RUUTU, Jussi**
FIN-02210 Espoo (FI)
• **LAKKAKORPI, Jani**
00200 Helsinki (FI)
• **RAISANEN, Vilho, I.**
FIN-00200 Helsinki (FI)

(74) Representative: **Ruuskanen, Juha-Pekka et al**
Page White & Farrer
Bedford House
John Street
London WC1N 2BF (GB)

(56) References cited:
EP-A- 1 206 087 US-A1- 2002 057 708
US-A1- 2002 122 428 US-A1- 2002 136 230
US-A1- 2003 028 663 US-A1- 2003 063 562
US-A1- 2003 118 016

- **Yoram Bernet: "Networking Quality of Service and Windows Operating Systems: Chapter 3: Queuing Mechanisms", , 14 November 2000 (2000-11-14), pages 1-24, XP055107735, Retrieved from the Internet: URL: <http://www.ablongman.com/samplechapter/1578702062.pdf> [retrieved on 2014-03-13]**
- **GIROUX N ET AL: "Queuing and Scheduling; Quality of service in ATM networks, Ch. 5", QUALITY OF SERVICE IN ATM NETWORKS: STATE-OF-THE-ART TRAFFIC, XX, XX, 1 January 1998 (1998-01-01), pages 85-121, XP002260612,**
- **BAKER CISCO SYSTEM K CHAN NORTEL NETWORKS A SMITH HARBOUR NETWORKS F: "Management Information Base for the Differentiated Services Architecture; rfc3289.txt", 5. JCT-VC MEETING; 96. MPEG MEETING; 16-3-2011 - 23-3-2011; GENEVA; (JOINT COLLABORATIVE TEAM ON VIDEO CODING OF ISO/IEC JTC1/SC29/WG11 AND ITU-T SG.16); URL: [HTTP://WFTP3.ITU.INT/AV-ARCH/JCTVC-SITE/](http://WFTP3.ITU.INT/AV-ARCH/JCTVC-SITE/), INTERNET ENGINEERING TASK FORCE, IETF, CH, 1 May 2002 (2002-05-01), XP015009067, ISSN: 0000-0003**
- **FLOYD S ET AL: "LINK-SHARING AND RESOURCE MANAGEMENT MODELS FOR PACKET NETWORKS", IEEE / ACM TRANSACTIONS ON NETWORKING, IEEE / ACM, NEW YORK, NY, US, vol. 3, no. 4, 1 August 1995 (1995-08-01), pages 365-386, XP000520857, ISSN: 1063-6692, DOI: 10.1109/90.413212**

Note: Within nine months of the publication of the mention of the grant of the European patent in the European Patent Bulletin, any person may give notice to the European Patent Office of opposition to that patent, in accordance with the Implementing Regulations. Notice of opposition shall not be deemed to have been filed until the opposition fee has been paid. (Art. 99(1) European Patent Convention).

EP 1 527 568 B1

Remarks:

The file contains technical information submitted after the application was filed and not included in this specification

Description

Field of the Invention

[0001] The present invention relates to apparatus for a programmable scheduler, to a router to a system, to a computer program, and to a method for programmable scheduling of routers. In particular, the present invention relates to IP networks and a programmable scheduling system for IP routers.

Background of the Invention

[0002] The introduction of Differentiated Services and other Quality of Service (QoS) methods has necessitated the deployment of several buffers per router output link. Traditionally, router vendors have implemented one, or possibly a few different ways to configure the scheduling between these different buffers for the router. However, this limits the possibilities offered by these technologies to vendor specific solutions and makes it difficult and expensive for operators to modify and update their routers.

[0003] Scheduling is an important part of any packet based data network element, such as IP routers. Scheduling is the functionality within the IP routers that decides which queue is served next. Effectively, scheduling decides from which buffer the next transmitted IP packet is picked, which may greatly affect performance.

[0004] Recently the increased demand for Quality of Service (QoS) has necessitated new solutions to replace the Best Effort paradigm. For example, Differentiated Services (DiffServ) is a framework standardized by the IETF that is based on the idea of differentiating IP traffic based on QoS requirements. In practice, this means multiple buffers per interface, so that, e.g., delay sensitive traffic can be handled through its own buffers.

[0005] The prior art solutions support narrow predefined scheduling schemes. Typically router vendors implement some simple basic schemes and may only allow very limited combinations of the predetermined schemes. This means that when an operator's requirements change, there is a need to upgrade the whole router software to support the change. This may involve, for example, more queues within the router. Effectively, the operator is closely tied with the scheduling model that the router vendor has selected.

[0006] From the vendor's point of view the same problem is seen as the need to redesign the scheduling part of IP routers each time that the standards or operators' requirement change. This means extra effort and costs not only for the operator but also for the vendor.

[0007] Thus, there is a need for a solution that allows a flexible way to configure the scheduling of IP routers without the need for updating the router software and/or hardware.

[0008] A known document, EP1206087A2, discloses a class of service manager that works in conjunction with a scheduler in order to provide policy based quality of

service. As data packets arrive in one or more of the class of service priority queues of a transaction FIFO (first in first out), the scheduler directs a selected packet pointer from one of the priority queues. The selection is based upon a queue scheduling algorithm, which is programmed by a user through a CPU, within the class of service manager.

[0009] GIROUX N ET AL: "Queuing and Scheduling; Quality of service in ATM networks, ch 5", XP002260612 is a chapter in a book that focuses on the design of queuing structures and scheduling algorithms at a contention point.

[0010] BAKER CISCO SYSTEM K CHAN NORTEL NETWORKS A SMITH HARBOUR NETWORKS: "Management Information Base for the Differentiated Services Architecture; rfc3289.txt", XP002260612 describes a Structure of Management Information v2 Management Information Base for a device implementing the Differentiated Services Architecture.

[0011] FLOYD S ET AL: "Link-sharing and resource management models for packet networks", XP000520857 relates to the use of link-sharing mechanisms in packet networks and presents algorithms for link-sharing.

Summary of the Invention

[0012] The present invention is directed at addressing the above-mentioned shortcomings, disadvantages and problems, and will be understood by reading and studying the following specification.

[0013] The present invention is defined by the appended independent claims. Certain more specific aspects are defined by the dependent claims.

[0014] Generally, a programmable scheduling system for IP routers is provided. According to the invention, a programmable scheduler for IP routers is provided that can support single stage and multistage scheduling. This allows flexible combinations of scheduling and increases the available QoS solutions available to operators.

[0015] According to the invention, the scheduling for the routers allows a flexible, programmable scheduling in one or many stages. With this kind of scheduling the router can be configured to support almost any known scheduling method or combination. Priority Queuing (PQ) and Deficit Round Robin (DRR) scheduling methods are used according to one embodiment of the invention.

[0016] According to the invention, the routers are programmed dynamically. Dynamic programming allows on the fly changes to the routers.

[0017] According to the invention, a Scheduler Configuration Interface is used to adjust the scheduling of the router. A set of scheduling parameters are used to define 1) the number and relative positions of the scheduling stages, 2) the scheduling method applied by each scheduling stage, 3) all relevant parameters for the selected method and 4) the position of the queues in the sched-

uling hierarchy. Accordingly, the programmable IP router is equipped with an interface that supports the above parameters.

Brief Description of the Drawings

[0018]

FIGURE 1 illustrates an exemplary mobile IP network in which the invention may operate;
FIGURE 2 shows a schematic diagram that illustrates an exemplary system overview in which local area networks and a wide area network are interconnected by routers;
FIGURE 3 illustrates a block diagram of an exemplary IP router;
FIGURE 4 illustrates queues with a multi-stage scheduling system;
FIGURE 5 illustrates a programmable scheduling system for an IP router;
FIGURE 6 illustrates a process for a programmable scheduling method; and,
FIGURE 7 illustrates how a general two-stage queuing model can be implemented with Programmable PQ/DRR Scheduler, in accordance with aspects of the invention.

Detailed Description of the Preferred Embodiment

[0019] In the following detailed description of exemplary embodiments of the invention, reference is made to the accompanied drawings, which form a part hereof, and which is shown by way of illustration, specific exemplary embodiments of which the invention may be practiced. Each embodiment is described in sufficient detail to enable those skilled in the art to practice the invention, and it is to be understood that other embodiments may be utilized, and other changes may be made, without departing from the scope of the present invention. The following detailed description is, therefore, not to be taken in a limiting sense, and the scope of the present invention is defined only by the appended claims.

[0020] Throughout the specification and claims, the following terms take the meanings explicitly associated herein, unless the context clearly dictates otherwise. The term "node" refers to a network element, such as a router. The term "importance" refers to a priority associated with a packet in terms of packet drop. The term "bandwidth broker" refers to a dedicated network element that knows the topology and link loads associated with the network. The term "flow" means a flow of packets. The term "urgency" refers to the packet transfer delay priority. The term support node refers to both "GGSN" and "SGSN" nodes. The term "user" refers to any person or customer such as a business or organization that employs a mobile device to communicate or access resources over a mobile network. The term "operator" refers to any technician or organization that maintains or services an IP based

network. "Priority Queuing" (PQ) means that the queue with the highest priority is always served first and the lower priority queues are served only when there are no packets in the higher priority queue(s). "Weighted Fair Queuing" (WFQ) is based on the idea that each queue is assigned a weight that defines how much service the queue receives. "Deficit Round Robin" (DRR) refers to a method where each queue is served periodically and a variable called a "quantum" tells how many bytes are drawn from the queue per turn. A "deficit counter" tells how many bytes the scheduler owes to particular flow. The term "virtual queue" refers to non-physical queues that are used to construct a multi-stage scheduling hierarchy. If all physical queues under a given virtual queue are empty, the virtual queue is removed from the scheduling hierarchy until a packet arrives to one of those physical queues that are under the virtual queue. The term "physical queue" means a real buffer.

[0021] Referring to the drawings, like numbers indicate like parts throughout the views. Additionally, a reference to the singular includes a reference to the plural unless otherwise stated or is inconsistent with the disclosure herein.

[0022] Briefly described, the present invention is directed at a programmable scheduling method for IP routers. A Programmable Scheduler can be re-configured dynamically to implement one stage or multi-stage scheduling through a Scheduler Configuration Interface. A set of parameters are used to define 1) the number and relative positions of the scheduling stages, 2) the scheduling method applied by each scheduling stage, 3) all relevant parameters for the selected method and 4) the position of the queues in the scheduling hierarchy. Accordingly, the router is equipped with an interface that supports the above parameters. According to one embodiment of the invention, the interface is called a Scheduling Configuration Interface. The programmable & hierarchical scheduler implementation can use various scheduling methods, such as Priority Queuing (PQ) or Deficit Round Robin (DRR) in any scheduling stage.

Illustrative Operating Environment

[0023] With reference to FIGURE 1, an exemplary mobile IP network in which the invention may operate is illustrated. As shown in the figure, mobile IP network 100 includes mobile node (MN) 105, radio access network (RAN) 110, SGSN 115, core network 120, routers 125_{A-F}, policy server 190, optional bandwidth broker (BB) 195, GGSNs 135_{A-B}, data network 140, and data network 145.

[0024] The connections and operation for mobile IP network 100 will now be described. MN 105 is coupled to radio access network (RAN) 110. Generally, MN 105 may include any device capable of connecting to a wireless network such as radio access network 110. Such devices include cellular telephones, smart phones, pagers, radio frequency (RF) devices, infrared (IR) devices, integrated devices combining one or more of the preced-

ing devices, and the like. MN 105 may also include other devices that have a wireless interface such as Personal Digital Assistants (PDAs), handheld computers, personal computers, multiprocessor systems, microprocessor-based or programmable consumer electronics, network PCs, wearable computers, and the like.

[0025] Radio Access Network (RAN) 110 manages the radio resources and provides the user with a mechanism to access core network 120. Radio access network 110 transports information to and from devices capable of wireless communication, such as MS 105. Radio access network 110 may include both wireless and wired components. For example, radio access network 110 may include a cellular tower that is linked to a wired telephone network. Typically, the cellular tower carries communication to and from cell phones, pagers, and other wireless devices, and the wired telephone network carries communication to regular phones, long-distance communication links, and the like. As shown in the figure, RAN 110 includes routers 125_{A-C}. Routers 125_{A-C} are programmable IP routers that may be configured to support almost any known scheduling method or combination. According to one embodiment of the invention, routers 125_{A-C} combine priority queuing (PQ) and deficit round robin (DRR). Bandwidth Broker (BB) 195, or some other network element, may modify the operation of programmable routers 125_{A-C} as a response to changing network conditions. These changing network conditions may include transfer delay, load and packet loss, and the like. The configuration of the routers may be a dynamic reconfiguration, even while on-line. Policy server 190, or some other dedicated network element, may be used to provide Quality of Service (QoS) rules relating to how the routers process the packets. Briefly described, policy server 190 may be used to monitor and aid in providing the appropriate behavior model for packet processing within the routers for IP based networks. According to one embodiment, each router may inform the bandwidth broker and the policy server of information relating to its operation and receive information from the bandwidth broker and the policy server to set the parameters and scheduling appropriately.

[0026] Some nodes may be General Packet Radio Service (GPRS) nodes. For example, Serving GPRS Support Node (SGSN) 115 may send and receive data from mobile nodes, such as MN 105, over RAN 110. SGSN 115 also maintains location information relating to MN 105. SGSN 115 communicates between MN 105 and Gateway GPRS Support Node (GGSN)s 135_{A-B} through core network 120. According to one embodiment of the invention, policy server 190 communicates with RAN 110 and core network 120.

[0027] Core network 120 is an IP packet based backbone network that includes routers, such as routers 125_{D-F}, to connect the support nodes in the network, SGSNs, GGSNs, a Policy Server and a Bandwidth Broker. Routers are intermediary devices on a communications network that expedite message delivery. On a single net-

work linking many computers through a mesh of possible connections, a router receives transmitted messages and forwards them to their correct destinations over available routes. Routers may be simple computing devices or complex computing devices. For example, a router may be a computer including memory, processors, and network interface units. According to one embodiment of the invention, routers 125_{D-F}, as 125_{A-C} are programmable IP routers that may be configured to support almost any known scheduling method or combination, such as PQ and DRR, as discussed above.

[0028] GGSNs 135_{A-B} are coupled to each other, to SGSNs and to external networks through routers 125_{A-C} and act as gateways to external networks, such as data network 140 and data network 145. Networks 140 and 145 may be the public Internet or a private data network. GGSNs 135_{A-B} allow MS 105 to access network 140 and network 145.

[0029] The operator may set QoS rules to determine whether or not to accept a packet based on different service classes for a particular user or group of users. For example, conversational traffic from user group A may be carried using standard EF behavior. The specific user of MN 105 may be differentiated into one of these user groups by an identifier associated with the user. For example, the identifier may be the user Mobile Station Integrated Services Digital Network (MSISDN) number that is known to both the SGSN and the GGSN support nodes.

[0030] Policy server 190 is coupled to core network 120 through communication mediums. Policy server 190 may be programmed by an operator with rules to manage the 3GPP quality-of-service (QoS) to IP QoS mapping for mobile IP network 100. More specifically, an operator may generate the rules that are used by the nodes on mobile IP network 100 to help ensure end-to-end QoS. These rules may be supplied to the nodes by policy server 190.

[0031] Bandwidth broker 195 is coupled to RAN 110 and core network 120 through communication mediums. BB 195 may be programmed by an operator with rules to manage, among other things, the scheduling parameters associated with the routers.

[0032] Furthermore, computers, and other related electronic devices may be connected to network 140 and network 145. The public Internet itself may be formed from a vast number of such interconnected networks, computers, and routers. Mobile IP network 100 may include many more components than those shown in FIGURE 1. However, the components shown are sufficient to disclose an illustrative embodiment for practicing the present invention.

[0033] The media used to transmit information in the communication links as described above illustrate one type of computer-readable media, namely communication media. Generally, computer-readable media includes any media that can be accessed by a computing device. Communication media typically embodies computer-readable instructions, data structures, program

modules, or other data in a modulated data signal such as a carrier wave or other transport mechanism and includes any information delivery media. The term "modulated data signal" means a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, communication media includes wired media such as twisted pair, coaxial cable, fiber optics, wave guides, and other wired media and wireless media such as acoustic, RF, infrared, and other wireless media.

[0034] FIGURE 2 shows another exemplary system in which the invention operates in which a number of local area networks ("LANs") 220_{a-d} and wide area network ("WAN") 230 interconnected by routers 210. On an interconnected set of LANs--including those based on differing architectures and protocols--, a router acts as a link between LANs, enabling messages to be sent from one to another.

[0035] Routers 210 are configured such that parameters may be dynamically set to show any desired scheduling scheme. According to one embodiment of the invention, routers 210 are programmable IP routers that may be adjusted to combine PQ and DRR scheduling functionality.

[0036] Communication links within LANs typically include twisted wire pair, fiber optics, or coaxial cable, while communication links between networks may utilize analog telephone lines, full or fractional dedicated digital lines including T1, T2, T3, and T4, Integrated Services Digital Networks (ISDNs), Digital Subscriber Lines (DSLs), wireless links, or other communications links. Furthermore, computers, such as remote computer 240, and other related electronic devices can be remotely connected to either LANs 220_{a-d} or WAN 230 via a modem and temporary telephone link. The number of WANs, LANs, and routers in FIGURE 2 may be increased or decreased without departing from the spirit or scope of this invention. As such, the Internet itself may be formed from a vast number of such interconnected networks, computers, and routers and that an embodiment of the invention could be practiced over the Internet without departing from the scope of the invention.

Programmable Scheduling for IP Routers

[0037] FIGURE 3 illustrates a block diagram of an exemplary IP router in accordance with aspects of the invention. As shown in the figure, IP router 300 includes DiffServ interface 310, ingress interface 320, QoS agent 330, scheduling configuration interface 355, routing core 340, and egress interface 350. Router 300 may include many more components than those shown in FIGURE 3. However, the components shown are sufficient to disclose an illustrative embodiment for practicing the present invention.

[0038] As illustrated in FIGURE 3, router 300 is divided into six functional blocks. Routing core 340 provides a router's normal routing and switching functionality. One

of the functionalities of the routing core is to select the egress interface for data packets entering through the ingress interface. The routing core moves packets between interfaces according to set policies. For purposes of this discussion, routing core 340 can be thought of as an infinite bandwidth, zero-delay backplane connecting interfaces.

[0039] DiffServ interface 310 monitors and provisions DiffServ operating parameters relating to the scheduling system. Monitored parameters may include statistics regarding traffic carried at various DiffServ service levels. These statistics may be used for accounting purposes and/or for tracking compliance to Traffic Conditioning Specifications (TCSs) negotiated with customers. Provisioned parameters are primarily the TCS parameters for Classifiers and Meters and the associated per-hop behavior (PHB) configuration parameters for actions and queuing elements. The operator, typically through a system administrator, interacts with DiffServ interface 310 through one or more management protocols, such as SNMP or COPS protocol, or through other router configuration tools. The operator may interact with DiffServ interface 310 through a bandwidth broker, policy server, a serial terminal, a telnet console, and the like.

[0040] Specific policy rules and goals governing the DiffServ behavior of a router are typically installed by policy management mechanisms.

[0041] Ingress interface 320, routing core 340 and egress interface 350 are illustrated at the center of the diagram illustrating router 300. In actual router implementations, there may be any number of ingress and egress interfaces interconnected by the routing core. The routing core element serves as an abstraction of a router's normal routing and switching functionality.

[0042] Optional QoS block 330 may be used to help enforce QoS requirements. Router 300 may snoop or participate in either per-microflow or per-flow-aggregate signaling of QoS requirements, e.g. using the RSVP protocol. Snooping of RSVP messages may be used, for example, to learn how to classify traffic without actually participating as a RSVP protocol peer. DiffServ router 300 may reject or admit RSVP reservation requests to provide a means of admission control to DiffServ-based services or it may use these requests to trigger provisioning changes for a flow-aggregation in the DiffServ network. If QoS block 330 is included in router 300, it may only be active in the control plane and not in the data plane. In this case, RSVP could be used merely to signal a reservation state without installing any actual reservations in the data plane of router 300. The data plane could still act purely on DiffServ DSCPs and provide PHBs for handling data traffic without the normal per-microflow handling expected to support some services.

[0043] Ingress interface 320 provides classification, metering, action, and queuing elements. Classifiers are parameterized by filters and output streams. Packets from the input stream are sorted into various output streams by filters which match the contents of the packet

or possibly match other attributes associated with the packet. The simplest classifier element is one that matches all packets that are applied at its input. In this case, the classifier may be omitted. A classifier may also classify the input streams according to their service class.

[0044] Ingress interface 320 may be configured to identify traffic into classes, meter it, and ensure that any excess traffic is appropriately dealt with according to the PHB. For example, for AF behavior, this may mean marking excess traffic; for EF, this may mean dropping excess traffic or shaping it to a maximum rate.

[0045] A packet arriving at ingress interface 320 picks up its policy from a classifier, which selects traffic according to some specification for each traffic class. Typically, a classifier identifies a variety of traffic and breaks it up into separate classes. According to one embodiment of the invention, packets are classified according to an importance priority and an urgency priority. More priorities may be implemented.

[0046] For network planning and perhaps other business purposes, such as for billing purposes, arriving traffic is normally counted. Therefore, a count action is configured. Also, traffic may be marked or remarked with the appropriate DSCP by a mark action. The operator may select the importance and urgency levels according to their business model. For instance, if the operator desires to maximize instant revenues, importance should have a direct relationship to the financial benefit of the packet to the network operator. When the objective of the operator is to support the primary business of the enterprise, preference could be given to those applications that are important to reach that objective..

[0047] Marking of the packet priorities is typically done in the first node that is controlled by the network operator. The packet typically does not need further marking by any of the other nodes. According to another embodiment of the invention, marking of the packet may be done in the mobile node. Under this scenario, a confidential relationship between the user and the operator is established. Packets may also be re-marked. For instance, the packets may be re-classified at the boundaries between network domains.

[0048] A metering element may be used to determine whether the stream is within a predetermined rate by monitoring real-time traffic conditions. As a result of the information obtained by the metering element actions may be triggered within the DiffServ router. For example, a metering element may be used to determine when the customer is being served according to their service level agreement (SLA). A meter measures the rate at which packets making up a stream or flow of traffic pass it, compares the measured rate to some set of thresholds and produces some number of potential results. A packet is said to conform to a specified level if, at the time that the packet is being examined, the stream appears to be within the rate limit for the profile associated with that level.

[0049] An action element sets the action to be applied to the packet. For example, the action element may mark

a packet, drop a packet according to the classification and queue in question, multiplex, count, or perform a null action. Actions are events in a DiffServ interface that may affect the packet. An action may calculate statistics on traffic in various configured classes, mark it with a DSCP, drop it, or queue it before passing it on for other processing.

[0050] DiffServ codepoint (DSCP) Markers are elements which set a codepoint (e.g. the DSCP in an IP header). The DSCP markings are typically applied either by a trusted upstream node, e.g. a customer, or by the edge routers on entry to the DiffServ network. DSCP Markers may also act on unmarked packets (e.g. those submitted with DSCP of zero) or may re-mark previously marked packets. The mark set in a packet will determine its subsequent PHB treatment in downstream nodes of a network and possibly also in subsequent processing stages within router 300. DSCP Markers for DiffServ are normally parameterized by a single parameter: the 6-bit DSCP to be marked in the packet header.

[0051] Egress interface 350 is arranged to receive data from DiffServ interface 310, routing core 340, scheduling configuration interface 355 and QoS block 330. Egress interface 350 also includes an output for outputting packets using the appropriate scheduling configuration. As mentioned above, many actions may be applied to the packets.

[0052] Queuing elements modulate the transmission of packets belonging to the different traffic streams and determine their ordering, possibly storing them temporarily or discarding them. Packets are usually stored either because there is a resource constraint (e.g., available bandwidth) which prevents immediate forwarding, or because the queuing block is being used to alter the temporal properties of a traffic stream (i.e. shaping). However, when several queues are used together in a queuing system, they can also achieve effects beyond that for given traffic streams. They can be used to limit variation in delay or impose a maximum rate (shaping), to permit several streams to share a link in a semi-predictable fashion, or to move variation in delay from some streams to other streams. The queuing elements in router 300 may be used to implement different behaviors, such as AF/EF behavior.

[0053] Scheduling configuration interface 355 includes parameters that set the scheduling of router 300. Generally, a scheduler is an element which gates the departure of each packet that arrives at one of its inputs, based on a service discipline. The scheduler typically has one or more inputs and one output. Each input has an upstream element to which it is connected, and a set of parameters that affects the scheduling of packets received at that input. The scheduler may utilize a variety of scheduling algorithms, including, but not limited to, priority queuing and deficit round robin. Other scheduling algorithms may also be used. Scheduling configuration interface uses a set of scheduling parameters to define 1) the number and relative positions of the scheduling

stages, 2) the scheduling method applied by each scheduling stage, 3) all relevant parameters for the selected method and 4) the position of the queues in the scheduling hierarchy. Schedulers may be used to shape traffic streams to match some profile by delaying packets that might be deemed non-conforming by some downstream node: a packet is delayed until such time as it would conform to a downstream meter using the same profile. PHBs, such as the class selectors, or AF/EF, have descriptions or configuration parameters which strongly suggest the sort of scheduling discipline needed to implement them.

[0054] FIGURE 4 illustrates queues with a multi-stage scheduling system, in accordance with aspects of the invention. As shown in the figure, system 400 includes scheduling 1 system 410, and scheduling 2 system 420.

[0055] In this exemplary case, the aim for the multi-stage scheduling system is for more controlled bandwidth share among traffic classes. This is especially needed at radio interfaces when traffic must be handled in a flow-based manner.

[0056] The combination of scheduling methods can be used with one-stage or multi-stage scheduling. In the latter case the stages can have completely different kind of scheduling methods. For example, the first stage (410), closest to the output link 425, can share the link bandwidth using weighted scheduling but the second stage scheduling (420) may use priority queuing.

[0057] FIGURE 5 illustrates a programmable scheduling system for an IP router, in accordance with aspects of the invention. As shown in the figure, scheduling system 500 includes programmable scheduling 510, set of scheduling parameters 502, output link 530, scheduling configuration interface 535, and queues 545.

[0058] According to the invention, programmable scheduling box 510 is a scheduler that can be re-configured dynamically to implement a one stage or multi-stage scheduling through Scheduler Configuration Interface 535. A set of scheduling parameters (520) are used to define 1) the number and relative positions of the scheduling stages, 2) the scheduling method applied by each scheduling stage, 3) all relevant parameters for the selected method and 4) the position of the queues in the scheduling hierarchy. Accordingly, the programmable IP router is equipped with an interface that supports the above parameters. According to the invention, the interface is called a scheduling configuration interface. According to the invention, the programmable and hierarchical scheduler implementation can use various scheduling methods, such as Priority Queueing or Deficit Round Robin, in any scheduling stage.

[0059] As illustrated in the figure, the router output consists of two or more physical queues. These queues can be implemented in hardware or software and their number N may be configurable. According to one embodiment of the invention, each physical queue is assigned a set of parameters including a stage parameter, a higher stage parameter, a method parameter, and

scheduling parameters. The stage parameter is the scheduling stage for the queue. The higher stage parameter refers to a higher stage to which the stage is connected. The method parameter refers to the scheduling method used in this stage. According to one embodiment of the invention, the scheduling method may be PQ or DRR. Other scheduling methods may be used. The scheduling parameters are the relevant parameters for the selected scheduling method. For example, serving order in PQ, or scheduling weight-quantum - in DRR.

[0060] The parameters are given as input for the Programmable PQ/DRR Scheduler that reconstructs the scheduling hierarchy and applies the parameters to obtain the desired scheduling method.

[0061] Effectively, for each physical queue a path through the scheduling hierarchy is defined. This defines, in an unambiguous way, the operation of the scheduler.

[0062] According to one embodiment of the invention, a linked list, starting at stage 1, is used to define the scheduling method. The link list is built according to the number of queues and their parameters (Stage, Higher stage, Method, Parameters).

[0063] According to one embodiment of the invention, the linked list of all physical and virtual queues is maintained according to the rules used in the original DRR algorithm. When a queue (virtual or physical) becomes empty, it is removed from the list. When a queue (virtual or physical) becomes non-empty, it is added to the tail of the list. When a DRR queue has been served (and the queue remains non-empty), it is first removed from the list and then added to the tail of the list.

[0064] This mechanism does not affect the Priority Queues (virtual or physical), since the serving order is marked to these queues. Since the queues carry their stage number, we can use the same list for all stages.

[0065] FIGURE 6 illustrates a process for a programmable scheduling method, in accordance with aspects of the invention. For purposes of explanation, the process flow is based on the idea that the router has two processes.

[0066] The first process (610) begins at block 660 at which point the packet arrives. Moving to block 665, the arriving IP packet is stored to the appropriate physical queue, based on, for example, the DiffServ Code Point (DSCP) at the IP packet header. After the packet arrival, and the storing of the packet to the appropriate queue, the process transitions to block 670 at which point the scheduling hierarchy is reconstructed (if needed).

[0067] The second process (605) starts from the highest level scheduler at block 615, which is connected to the output link. Moving to block 620, the scheduler selects an input based on the scheduling method. According to one embodiment of the invention, the scheduling method may be a combination of PQ and DRR. Transitioning to block 625, if needed, the scheduler parameters are updated (deficits are subtracted by quantum in the case of DRR). Flowing to decision block 630 a determination is made as to whether the input is connected to a virtual

queue (instead of a physical one). When it is, the process flows to block 635 where the process is repeated at the next stage. When the input is not connected to a virtual queue, then the physical queue is found and the process moves to block 640. At block 640, the packet is taken from the queue and transmitted to output link. Next, at block 645, the scheduler parameters may be updated if needed.

[0068] FIGURE 7 illustrates how a general two-stage queuing model can be implemented with Programmable PQ/DRR Scheduler, in accordance with aspects of the invention. As shown in the figure, queuing model 700 includes stage 710, stage 720, stage 730, stage 740, and stage 750.

[0069] Stage 750 is coupled to three queues (751-753) and uses weighted fair queuing. Stage 740 is coupled to queues 741-743 and uses priority queuing. Stage 730 is coupled to queues 731-733 and uses DRR, wherein the scheduling rates (quantums) are 50, 30, and 20, respectively for the three queues. Stage 720 is coupled to the output of stage 730 and queue 721 and uses PQ and has a serving order of 1 and 2, respectively, for the queues. Stage 710 is coupled to queue 711 and the output of queue 720 and uses PQ with scheduling weights of 1 and 2, respectively.

[0070] The present invention provides many advantages. One advantage is that the same implementation of the router product can be extremely future-proof in terms of scheduling. When an operator's requirements change, no new implementation is needed. This reduces R&D costs for routers. In addition, this kind of solution can be expected to be popular among operators. The risk for an operator to select the wrong QoS approach for IP routers is heavily reduced.

[0071] Another advantage is that there are no restrictions for scheduling the router, such as only being able to choose from a small set of predefined scheduling methods and make some simple combinations, typically only on one stage.

[0072] The above specification, examples and data provide a complete description of the manufacture and use of the composition of the invention. Since many embodiments of the invention can be made, the invention resides in the claims hereinafter appended.

Claims

1. An apparatus for a programmable scheduler, comprising:

a scheduling configuration interface (355, 535) coupled to an interface arrangement (320, 350), the interface arrangement being for receiving and outputting packets, whereas the scheduling configuration interface (355, 535) is further coupled to a routing core (340), wherein the routing core is for providing routing

and switching functionality relating to the packets and the interface arrangement (320, 350) is for coupling to the routing core (340);

the scheduling configuration interface (355, 535) is arranged to provide a scheduling configuration, and is programmed to implement a number of scheduling algorithms such that the interface arrangement (320, 350) is able to output packets according to the scheduling configuration;

the apparatus is for a programmable scheduler that can dynamically on the fly be reconfigured to support single stage and multi-stage scheduling (410, 420) through the scheduling configuration interface; and

wherein parameters relating to the scheduling configuration are received at the scheduling configuration interface (355, 535) to be used to determine the scheduling configuration, the parameters defining a number and relative positions of scheduling stages, a scheduling method applied by the scheduling stages, parameters for the scheduling method, and a position of queues in a scheduling hierarchy.

2. The apparatus of claim 1, wherein the scheduling configuration uses a Priority Queuing method and/or a Deficit Round Robin method in any scheduling stage.
3. The apparatus of any one of the preceding Claims, wherein the interface arrangement (320, 250) is further configured to classify the packet into separate classes.
4. The apparatus of Claim 3, further comprising a Quality of Service block coupled to the interface arrangement (320, 250), the scheduling configuration interface (355, 535) being configured to enforce Quality of Service requirements.
5. The apparatus of any one of the preceding claims, wherein the programmable scheduler is for an internet protocol router.
6. A router (300) comprising:

an interface arrangement (320,350) arranged to receive the packets and transmit packets to nodes across a network;

a scheduling configuration interface (355, 535) coupled to a routing core (340), wherein the routing core is for providing routing and switching functionality relating to the packets and the interface arrangement (320, 350) is for coupling to the routing core (340), the scheduling configuration interface being configured to receive parameters (520) relating to a scheduling method;

- a programmable scheduling component (510) configured to implement a one stage scheduling method and a multi-stage scheduling method (410, 420) in response to the parameters (520) received by the scheduling configuration interface (355, 535), wherein the programmable scheduling component (510) can be reconfigured dynamically on the fly to implement a one stage scheduling method and a multi-stage scheduling method through the scheduling configuration interface in response to parameters received by the scheduling configuration interface (355, 535), wherein the parameters received by the scheduling configuration interface define a number and relative positions of scheduling stages, a scheduling method applied by the scheduling stages, parameters for the scheduling method, and a position of queues in a scheduling hierarchy; and
an output (530) arranged to output packets such that the scheduling method is followed.
7. The router (300) of Claim 6, wherein the programmable scheduling component receives input from a plurality of scheduling methods.
 8. The router (300) of Claim 7, wherein the plurality of scheduling methods may use at least two queues.
 9. The router (300) of Claim 6, wherein the queues may be implemented in hardware.
 10. The router (300) of Claim 6, wherein the queues may be implemented in software.
 11. The router (300) of Claim 6, wherein the queues may be implemented in hardware and software.
 12. The router (300) of any of Claims 7 to 8, wherein the plurality of scheduling methods comprise a Priority Queuing method and a Deficit Round Robin method.
 13. The router (300) of any of Claims 6 to 12, wherein the parameters received by the scheduling configuration interface further comprise scheduling methods applied by scheduling stages.
 14. The router (300) of any of Claims 6 to 13, wherein the parameters received by the scheduling configuration interface further comprise parameters relating to a selected scheduling method.
 15. The router (300) of any one of claims 6 to 14, wherein said router is an internet protocol router.
 16. A system (500) for a programmable scheduler relating to internet protocol routers, comprising:
 - nodes, wherein the nodes include an interface arrangement (320, 350) arranged to communicate packets across the network; and
 - a router (300) according to any one of claims 6 to 15.
 17. A method for a programmable scheduler supporting single stage and multi-stage scheduling, comprising:
 - receiving and outputting packets by an interface arrangement (320, 350);
 - routing and switching the packets by a routing core (340) coupled to the interface arrangement (320, 350) and a scheduling configuration interface (355);
 - providing, by the scheduling configuration interface (355), a scheduling configuration and implementing a number of scheduling algorithms such that the interface arrangement (320, 350) is able to output packets according to the scheduling configuration;
 - receiving parameters at the scheduling configuration interface (355, 535), wherein the programmable scheduler can be reconfigured dynamically on the fly to implement the one stage scheduling and the multi-stage scheduling through the scheduling configuration interface in response to parameters received by the scheduling configuration interface (355, 535); and
 - determining the scheduling configuration based on the received parameters, the parameters defining a number and relative positions of scheduling stages, a scheduling method applied by the scheduling stages, parameters for the scheduling method and a position of queues in a scheduling hierarchy.
 18. The method of Claim 17, further comprising:
 - storing the received packets in a physical queue;
 - determining when to reconstruct a scheduling hierarchy, and when, reconstructing the scheduling hierarchy;
 - selecting a scheduling method;
 - selecting an input based on the scheduling method;
 - determining when to update scheduler parameters;
 - determining when the input is connected to a virtual queue, and when, proceeding until the input is connected to the physical queue, otherwise, taking a received packet from the physical queue and transmitting the packet to an output link.
 19. The method of Claim 18, wherein determining when to reconstruct the scheduling hierarchy further com-

prises determining when parameters have been received that relate to the scheduling hierarchy.

20. The method of Claim 18, further comprising dynamically receiving the parameters, and wherein selecting the scheduling method further comprises programming the scheduling method in relation to the dynamically received parameters. 5
21. The method of any of Claims 17 to 20, comprising using a Priority Queuing method and/or a Deficit Round Robin method in any of the scheduling stages. 10
22. The method of any of Claims 17 to 21, wherein the method is for programmable scheduling of internet protocol routers (300). 15
23. A computer program comprising program code means, which, when executed on a computer, cause the computer to perform the method steps of any one of claims 17 to 22. 20

Patentansprüche 25

1. Vorrichtung für ein programmierbares Steuerprogramm, umfassend:
 - eine Steuerkonfigurationsschnittstelle (355, 535), die mit einer Schnittstellenanordnung (320, 350) gekoppelt ist, wobei die Schnittstellenanordnung für das Empfangen und Ausgeben von Paketen ist, während die Steuerkonfigurationsschnittstelle (355, 535) ferner mit einem Routingkern (340) gekoppelt ist, wobei der Routingkern für das Bereitstellen von Routing- und Schaltfunktionalität in Bezug auf die Pakete ist und die Schnittstellenanordnung (320, 350) für das Koppeln mit dem Routingkern (340) ist; die Steuerkonfigurationsschnittstelle (355, 535) ist angeordnet, um eine Steuerkonfiguration bereitzustellen, und programmiert, eine Anzahl von Steueralgorithmus derart bereitzustellen, dass die Schnittstellenanordnung (320, 350) fähig ist, Pakete gemäß der Steuerkonfiguration auszugeben; 30
 - die Vorrichtung ist für ein programmierbares Steuerprogramm, das dynamisch im laufenden Betrieb umkonfiguriert werden kann, um Einzelphasen- und Mehrphasensteuern (410, 420) durch die Steuerkonfigurationsschnittstelle zu unterstützen; und 35
 - wobei Parameter in Bezug auf die Steuerkonfiguration an der Steuerkonfigurationsschnittstelle (355, 525) empfangen werden, um verwendet zu werden, die Steuerkonfiguration festzustellen, wobei die Parameter eine Anzahl und rela- 40

tive Positionen von Steuerphasen, ein Steuerungsverfahren, das von den Steuerphasen angewendet wird, Parameter für das Steuerungsverfahren und eine Position von Warteschlangen in einer Steuerhierarchie definieren.

2. Vorrichtung nach Anspruch 1, wobei die Steuerkonfiguration ein Prioritätswarteschlangenverfahren und/oder ein Defizit-"Round Robin"-Verfahren in einer Steuerphase verwendet. 45
3. Vorrichtung nach einem der vorstehenden Ansprüche, wobei die Schnittstellenanordnung (320, 350) ferner konfiguriert ist, das Paket in separate Klassen zu klassifizieren. 50
4. Vorrichtung nach Anspruch 3, ferner einen Servicequalitätsblock umfassend, der mit der Schnittstellenanordnung (320, 250) gekoppelt ist, wobei die Steuerkonfigurationsschnittstelle (355, 535) konfiguriert ist, Servicequalitätserfordernisse zu erzwingen.
5. Vorrichtung nach einem der vorstehenden Ansprüche, wobei das programmierbare Steuerprogramm für einen Internet-Protokoll-Router ist.
6. Ein Router (300), umfassend: 55

eine Schnittstellenanordnung (320, 350), die angeordnet ist, um die Pakete zu empfangen und die Pakete an Knoten in einem Netzwerk zu senden;

eine Steuerkonfigurationsschnittstelle (355, 535), die mit einem Routingkern (340) gekoppelt ist, wobei der Routingkern für das Bereitstellen von Routing und Schaltfunktionalität in Bezug auf die Pakete ist und die Schnittstellenanordnung (320, 350) für das Koppeln mit dem Routingkern (340) ist, wobei die Steuerkonfigurationsschnittstelle konfiguriert ist, Parameter (520) in Bezug auf ein Steuerungsverfahren zu empfangen;

eine programmierbare Steuerkomponente (510), die konfiguriert ist, ein Einphasensteuerungsverfahren und ein Mehrphasensteuerungsverfahren (410, 420) in Reaktion auf die Parameter (520), die von der Steuerkonfigurationsschnittstelle (355, 535) empfangen wurden, zu implementieren, wobei die programmierbare Steuerkomponente (510) dynamisch im laufenden Betrieb umkonfiguriert werden kann, um ein Einphasensteuerungsverfahren und ein Mehrphasensteuerungsverfahren durch die Steuerkonfigurationsschnittstelle in Reaktion auf Parameter, die von der Steuerkonfigurationsschnittstelle (355, 535) empfangen wurden, zu implementieren, wobei die von der Steuerkonfigurationsschnittstelle empfangenen Parameter eine Anzahl und rela-

- tive Positionen von Steuerphasen, ein Steuer-
verfahren, das von den Steuerphasen ange-
wendet wird, Parameter für das Steuerverfahren
und eine Position von Warteschlangen in einer
Steuerhierarchie definieren; und
eine Ausgabe (530), die angeordnet ist, um Pa-
kete derart auszugeben, dass dem Steuerver-
fahren gefolgt wird.
7. Router (300) nach Anspruch 6, wobei die program-
mierbare Steuerkomponente Eingaben von einer
Vielzahl von Steuerverfahren empfängt.
8. Router (300) nach Anspruch 7, wobei die Vielzahl
der Steuerverfahren mindestens zwei Warteschlan-
gen verwenden kann.
9. Router (300) nach Anspruch 6, wobei die Warte-
schlangen in Hardware implementiert sein können.
10. Router (300) nach Anspruch 6, wobei die Warte-
schlangen in Software implementiert sein können.
11. Router (300) nach Anspruch 6, wobei die Warte-
schlangen in Hardware und Software implementiert
sein können.
12. Router (300) nach einem der Ansprüche 7 bis 8, wo-
bei die Vielzahl der Steuerverfahren ein Prioritäts-
warteschlangenverfahren und ein Defizit-"Round
Robin"-Verfahren umfassen.
13. Router (300) nach einem der Ansprüche 6 bis 12,
wobei die Parameter, die von der Steuerkonfigura-
tionsschnittstelle empfangen werden, ferner Steuer-
verfahren umfassen, die von Steuerphasen ange-
wendet werden.
14. Router (300) nach einem der Ansprüche 6 bis 13,
wobei die Parameter, die von der Steuerkonfigura-
tionsschnittstelle empfangen werden, ferner Para-
meter in Bezug auf ein ausgewähltes Steuerverfah-
ren umfassen.
15. Router (300) nach einem der Ansprüche 6 bis 14,
wobei der Router ein Internet-Protokoll-Router ist.
16. System (500) für ein programmierbares Steuerpro-
gramm in Bezug auf Internet-Protokoll-Router, um-
fassend:
- Knoten, wobei die Knoten eine Schnittstellena-
nordnung (320, 350) umfassen, die angeordnet
sind, um Pakete über das Netzwerk zu kommu-
nizieren; und
ein Router (300) nach einem der Ansprüche 6
bis 15.

17. Verfahren für ein programmierbares Steuerpro-
gramm, das Einzelphasen- und Mehrphasensteuern
unterstützt, umfassend:

Empfangen und Ausgeben von Paketen durch
eine Schnittstellenanordnung (320, 350);
Routing und Schalten der Pakete von einem
Routingkern (340), der mit der Schnittstellena-
nordnung (320, 350) und einer Steuerkonfigu-
rationsschnittstelle (355) verbunden ist;
Bereitstellen, durch die Steuerkonfigurations-
schnittstelle (355) einer Steuerkonfiguration
und Implementieren einer Anzahl von Steuer-
algorithmen derart, dass die Schnittstellena-
nordnung (320, 350) fähig ist, Pakete gemäß der
Steuerkonfiguration auszugeben;
Empfangen von Parametern an der Steuerkon-
figurationsschnittstelle (355, 535), wobei das
programmierbare Steuerprogramm dynamisch
im laufenden Betrieb umkonfigurierbar ist, um
das Einphasensteuern und das Mehrphasen-
steuern durch die Steuerkonfigurationsschnitt-
stelle in Reaktion auf Parameter, die von der
Steuerkonfigurationsschnittstelle (355, 535)
empfangen werden, zu implementieren; und
Feststellen der Steuerkonfiguration auf Basis
der empfangenen Parameter, wobei die Para-
meter eine Anzahl und relative Positionen von
Steuerphasen, ein Steuerverfahren, das von
den Steuerphasen angewendet wird, Parame-
ter für das Steuerverfahren und eine Position
von Warteschlangen in einer Steuerhierarchie
definieren.

18. Verfahren nach Anspruch 17, ferner umfassend:

Speichern der empfangenen Pakete in einer
physikalischen Warteschlange;
Feststellen, wann eine Steuerhierarchie rekon-
struiert werden soll, und wenn die Steuerhierar-
chie rekonstruiert wird;
Auswählen eines Steuerverfahrens;
Auswählen einer Eingabe auf Basis des Steu-
erverfahrens;
Feststellen, wann Steuerprogrammparameter
zu aktualisieren sind;
Feststellen, wann die Eingabe mit einer virtuel-
len Warteschlange verbunden ist, und wann das
Fortfahren bis die Eingabe mit der physikali-
schen Warteschlange verbunden ist, ansons-
ten, Nehmen eines empfangenen Pakets aus
der physikalischen Warteschlange und Senden
des Pakets an einen Ausganglink.

19. Verfahren nach Anspruch 18, wobei das Feststellen,
wann die Steuerhierarchie zu rekonstruieren ist, fer-
ner das Feststellen umfasst, wann Parameter, die
sich auf die Steuerhierarchie beziehen, empfangen

wurden.

20. Verfahren nach Anspruch 18, ferner dynamisches Empfangen der Parameter umfassend, und wobei das Auswählen des Steuerverfahrens ferner Programmieren des Steuerverfahrens in Bezug auf die dynamisch empfangenen Parameter umfasst. 5
21. Verfahren nach einem der Ansprüche 17 bis 20, das Verwenden eines Prioritätswarteschlangenverfahrens und/oder eines Defizit-"Round Robin"-Verfahrens in einer der Steuerphasen umfassend. 10
22. Verfahren nach einem der Ansprüche 17 bis 21, wobei das Verfahren für programmierbares Steuern von Internet-Protokoll-Routern (300) ist. 15
23. Computerprogramm, Programmcodemittel umfassend, die, wenn sie auf einem Computer ausgeführt werden, den Computer veranlassen, die Verfahrensschritte nach einem der Ansprüche 17 bis 22 auszuführen. 20

Revendications 25

1. Appareil pour un ordonnanceur programmable, comprenant :

une interface de configuration d'ordonnement (355, 535) couplée à un agencement d'interface (320, 350), l'agencement d'interface étant destiné à recevoir et à transmettre des paquets, tandis que l'interface de configuration d'ordonnement (355, 535) est en outre couplée à un coeur de routage (340), dans lequel le coeur de routage est destiné à fournir une fonctionnalité de routage et de commutation se rapportant aux paquets et l'agencement d'interface (320, 350) est destinée à être couplée au coeur de routage (340) ; l'interface de configuration d'ordonnement (355, 535) est conçue pour fournir une configuration d'ordonnement et est programmée pour mettre en oeuvre un certain nombre d'algorithmes d'ordonnement de telle sorte que l'agencement d'interface (320, 350) puisse transmettre des paquets en fonction de la configuration d'ordonnement ; l'appareil est destiné à un ordonnanceur programmable qui peut de manière dynamique être reconfiguré au vol pour prendre en charge un ordonnement à une étape unique et à étapes multiples (410, 420) au moyen de l'interface de configuration d'ordonnement ; et dans lequel des paramètres se rapportant à la configuration d'ordonnement sont reçus au

niveau de l'interface de configuration d'ordonnement (355, 535) à utiliser pour déterminer la configuration d'ordonnement, les paramètres définissant un nombre, et des positions relatives, d'étapes d'ordonnement, un procédé d'ordonnement appliqué par les étapes d'ordonnement, des paramètres pour le procédé d'ordonnement et une position de files d'attente dans une hiérarchie d'ordonnement.

2. Appareil selon la revendication 1, dans lequel la configuration d'ordonnement utilise un procédé de mise en file d'attente à priorité et/ou un procédé de déficit Round-Robin dans n'importe quelle étape d'ordonnement.
3. Appareil selon l'une quelconque des revendications précédentes, dans lequel l'agencement d'interface (320, 250) est en outre configuré pour classer le paquet en classes distinctes.
4. Appareil selon la revendication 3, comprenant en outre un bloc de qualité de service couplé à l'agencement d'interface (320, 250), l'interface de configuration d'ordonnement (355, 535) étant configurée pour renforcer les exigences de qualité de service.
5. Appareil selon l'une quelconque des revendications précédentes, dans lequel l'ordonneur programmable est destiné à un routeur de protocole Internet.
6. Routeur (300) comprenant :
 - un agencement d'interface (320, 350) conçu pour recevoir le paquet et pour transmettre des paquets à des noeuds à travers un réseau ;
 - une interface de configuration d'ordonnement (355, 535) couplée à un coeur de routage (340), dans lequel le coeur de routage est destiné à fournir une fonctionnalité de routage et de commutation se rapportant aux paquets et l'agencement d'interface (320, 350) est destinée à être couplée au coeur de routage (340), l'interface de configuration d'ordonnement étant configurée pour recevoir des paramètres (520) se rapportant à un procédé d'ordonnement ;
 - un composant d'ordonnement programmable (510) configuré pour mettre en oeuvre un procédé d'ordonnement à une étape et un procédé d'ordonnements à étapes multiples (410, 420) en réponse à des paramètres (520) reçus par l'interface de configuration d'ordonnement (355, 535), dans lequel le composant d'ordonnement programmable (510) c'était comment configurer de manière dynami-

- que au vol pour mettre en oeuvre un procédé d'ordonnancement à une étape et un procédé d'ordonnancement à étapes multiples au moyen de l'interface de configuration d'ordonnancement en réponse à des paramètres reçus par l'interface de configuration d'ordonnancement (355, 535), dans lequel les paramètres reçus par l'interface de configuration d'ordonnancement définissent un nombre, et des positions relatives, d'étapes d'ordonnancement, un procédé d'ordonnancement appliqué par les étapes d'ordonnancement, des paramètres pour le procédé d'ordonnancement et une position de files d'attente dans une hiérarchie d'ordonnancement ; et
une sortie (530) conçu pour transmettre des paquets de telle sorte que le procédé d'ordonnancement soit suivi.
7. Routeur (300) selon la revendication 6, dans lequel le composant d'ordonnancement programmable reçoit une entrée en provenance d'une pluralité de procédés d'ordonnancement.
 8. Routeur (300) selon la revendication 7, dans lequel la pluralité de procédés d'ordonnancement peut utiliser au moins deux files d'attente.
 9. Routeur (300) selon la revendication 6, dans lequel les files d'attente peuvent être mises en oeuvre dans un matériel.
 10. Routeur (300) selon la revendication 6, dans lequel les files d'attente peuvent être mises en oeuvre dans un logiciel.
 11. Routeur (300) selon la revendication 6, dans lequel les files d'attente peuvent être mises en oeuvre dans un matériel et un logiciel.
 12. Routeur (300) selon l'une quelconque des revendications 7 à 8, dans lequel la pluralité de procédés d'ordonnancement comprennent un procédé de mise en file d'attente à priorité et un procédé de déficit Round-Robin.
 13. Routeur (300) selon l'une quelconque des revendications 6 à 12, dans lequel les paramètres reçus par l'interface de configuration d'ordonnancement comprennent en outre des procédés d'ordonnancement appliqués par des étapes d'ordonnancement.
 14. Routeur (300) selon l'une quelconque des revendications 6 à 13, dans lequel les paramètres reçus par l'interface de configuration d'ordonnancement comprennent en outre des paramètres se rapportant à un procédé d'ordonnancement sélectionné.
 15. Routeur (300) selon l'une quelconque des revendications 6 à 14, dans lequel ledit routeur est un routeur de protocole Internet.
 16. Système (500) pour un ordonnanceur programmable se rapportant à des routeurs de protocole Internet, comprenant :
des noeuds, dans lequel les noeuds comprennent un agencement d'interface (320, 350) conçu pour communiquer des paquets à travers le réseau ; et
un routeur (300) selon l'une quelconque des revendications 6 à 15.
 17. Procédé pour un ordonnanceur programmable prenant en charge un ordonnancement à une seule étape et à étapes multiples, consistant en :
à recevoir et à transmettre des paquets au moyen d'un agencement d'interface (320, 350) ;
à acheminer et à commuter les paquets au moyen d'un coeur de réseau (340) couplé à l'agencement d'interface (320, 350) et à une interface de configuration d'agencement (355) ;
à fournir, au moyen de l'interface de configuration d'ordonnancement (355), une configuration d'ordonnancement et à mettre en oeuvre un certain nombre d'algorithmes d'ordonnancement de telle sorte que l'agencement d'interface (320, 350) puisse transmettre des paquets en fonction de la configuration d'ordonnancement ;
à recevoir des paramètres au niveau de l'interface de configuration d'ordonnancement (355, 535), dans lequel l'ordonnanceur programmable peut être configuré pour mettre en oeuvre de manière dynamique au vol l'ordonnancement à une seule étape et l'ordonnancement à étapes multiples au moyen de l'interface de configuration d'ordonnancement en réponse à des paramètres reçus par l'interface de configuration d'ordonnancement (355 ; 535) ; et
à déterminer la configuration d'ordonnancement en se basant sur les paramètres reçus, les paramètres définissant un nombre, et des positions relatives, d'étapes d'ordonnancement, un procédé d'ordonnancement appliqué par les étapes d'ordonnancement, des paramètres pour le procédé d'ordonnancement et une position de files d'attente dans une hiérarchie d'ordonnancement.
 18. Procédé selon la revendication 17, consistant en outre :
à stocker les paquets reçus dans une file d'attente physique ;
à déterminer lorsqu'il faut reconstruire une hié-

- rarchie d'ordonnancement, et lorsqu'il le faut, à reconstruire la hiérarchie d'ordonnancement ;
à sélectionner un procédé d'ordonnancement ;
à sélectionner une entrée basée sur le procédé d'ordonnancement ; 5
à déterminer lorsqu'il faut mettre à jour des paramètres d'ordonnancement ;
à déterminer lorsque l'entrée est raccordée à une file d'attente virtuelle et, lorsque c'est le cas, à continuer jusqu'à ce que l'entrée soit raccordée à la file d'attente physique ; sinon, à prélever un paquet reçu de la file d'attente physique et à transmettre le paquet à une liaison de sortie. 10
- 19.** Procédé selon la revendication 18, dans lequel la détermination lorsqu'il faut reconstruire la hiérarchie d'ordonnancement consiste en outre à déterminer lorsque des paramètres ont été reçus qui se rapportent à la hiérarchie d'ordonnancement. 15
- 20.** Procédé selon la revendication 18, consistant en outre à recevoir de manière dynamique les paramètres et dans lequel la sélection du procédé d'ordonnancement consiste en outre à programmer le procédé d'ordonnancement par rapport aux paramètres reçus de manière dynamique. 20 25
- 21.** Procédé selon l'une quelconque des revendications 17 à 20, consistant à utiliser un procédé de mise en file d'attente à priorité et/ou un procédé de déficit Round-Robin dans l'une quelconque des étapes d'ordonnancement. 30
- 22.** Procédé selon l'une quelconque des revendications 17 à 21, dans lequel le procédé est destiné à un ordonnancement programmable de routeurs de protocole Internet (300). 35
- 23.** Programme d'ordinateur comprenant des moyens de code de programme qui, lorsqu'ils sont exécutés sur un ordinateur, contraignent l'ordinateur à réaliser les étapes du procédé selon l'une quelconque des revendications 17 à 22. 40

45

50

55

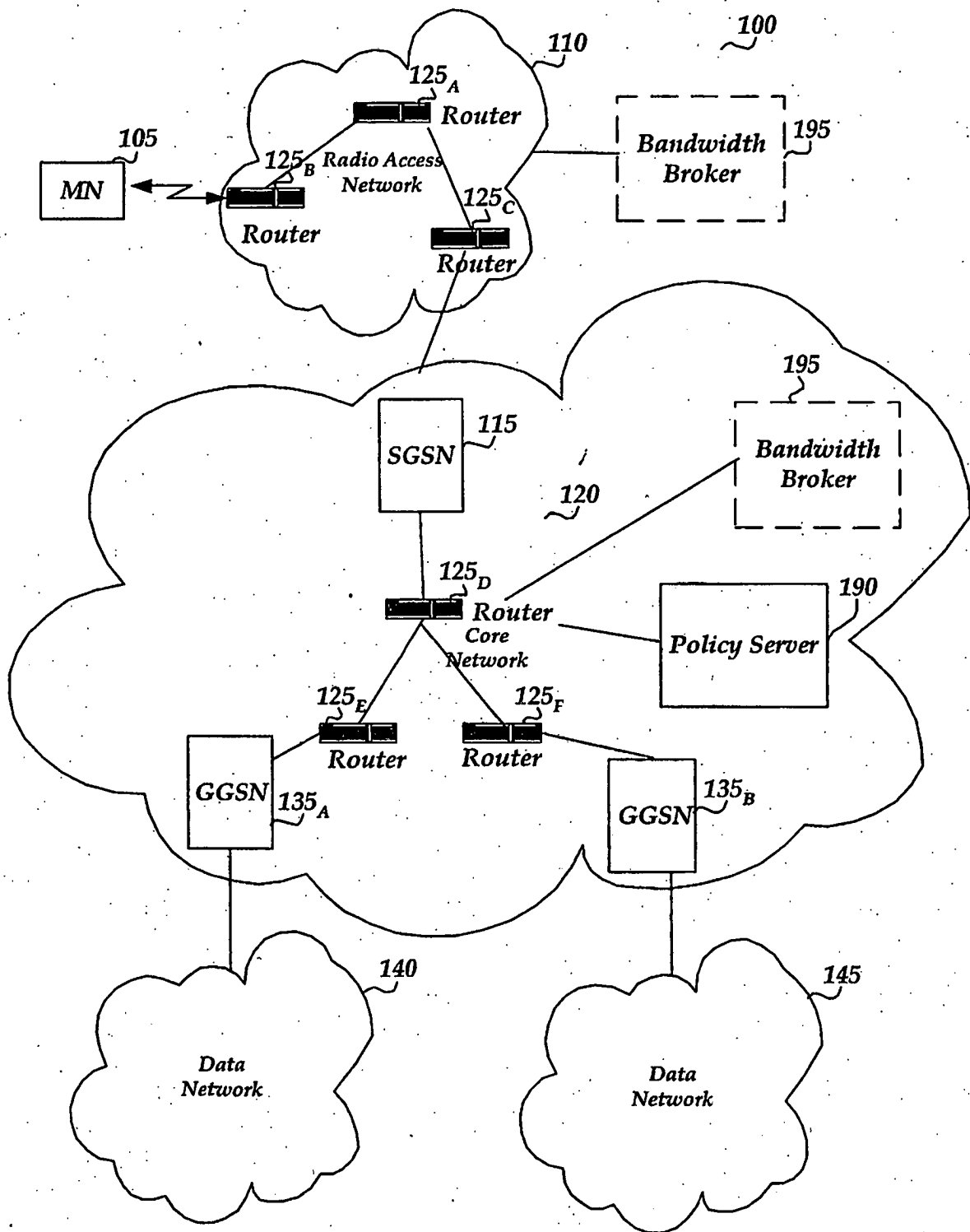


Fig.1

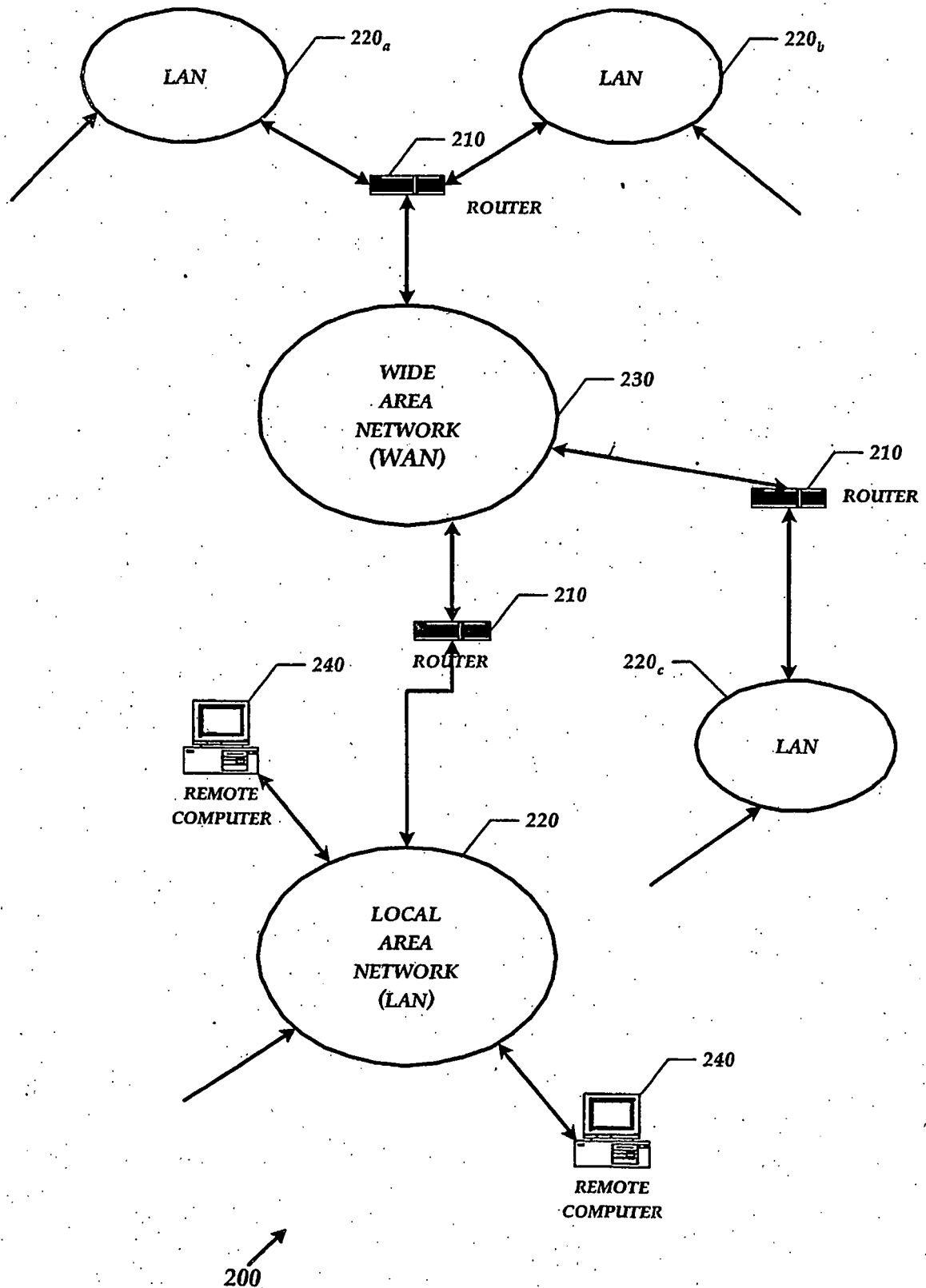


Fig.2

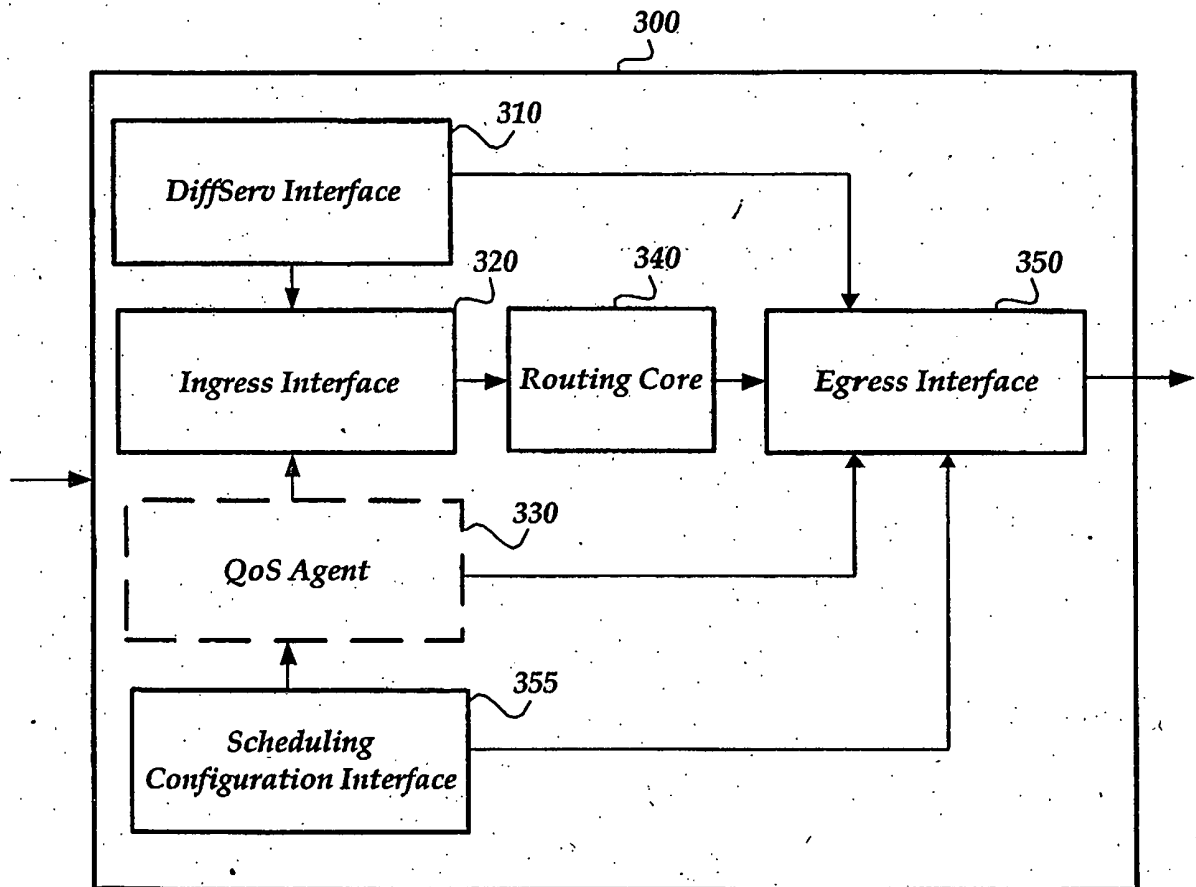


Fig.3

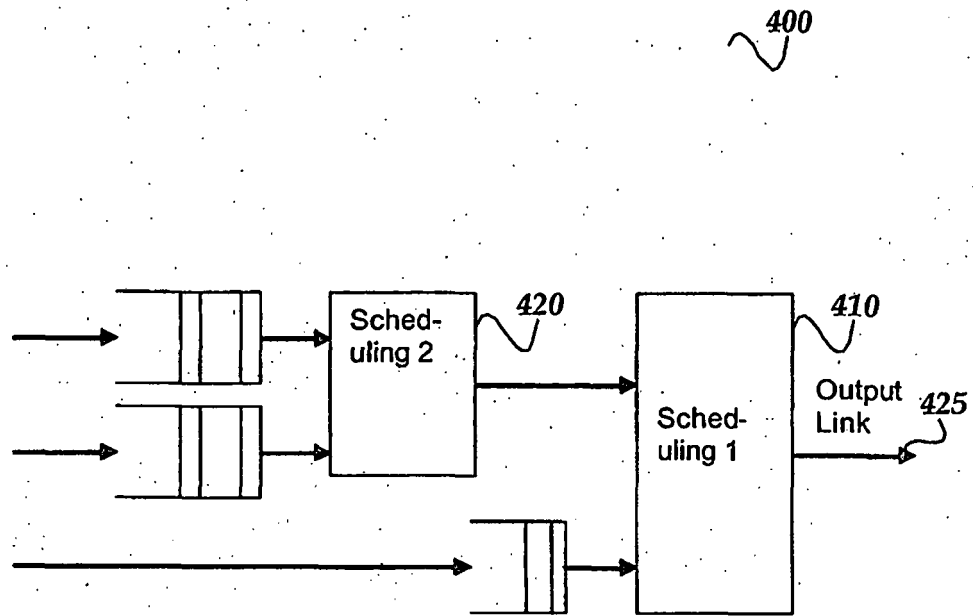


Fig.4

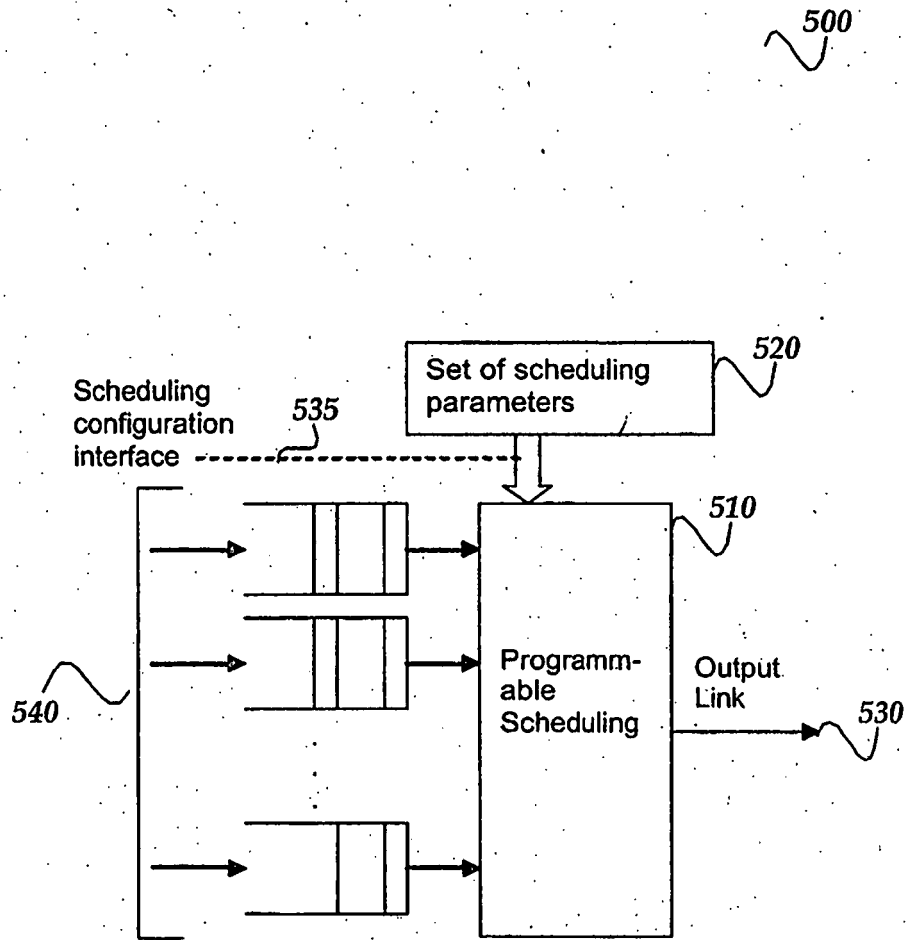
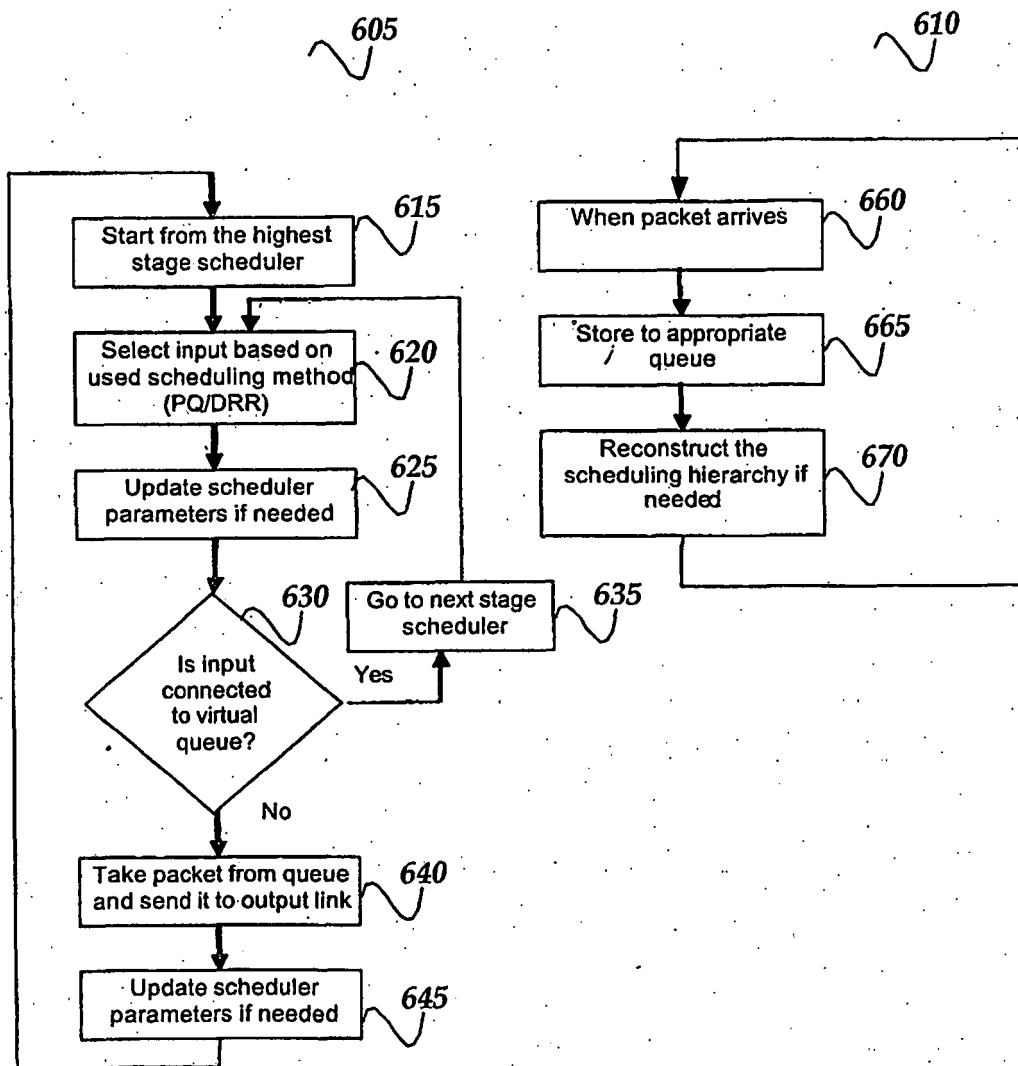


Fig.5

**Fig.6**

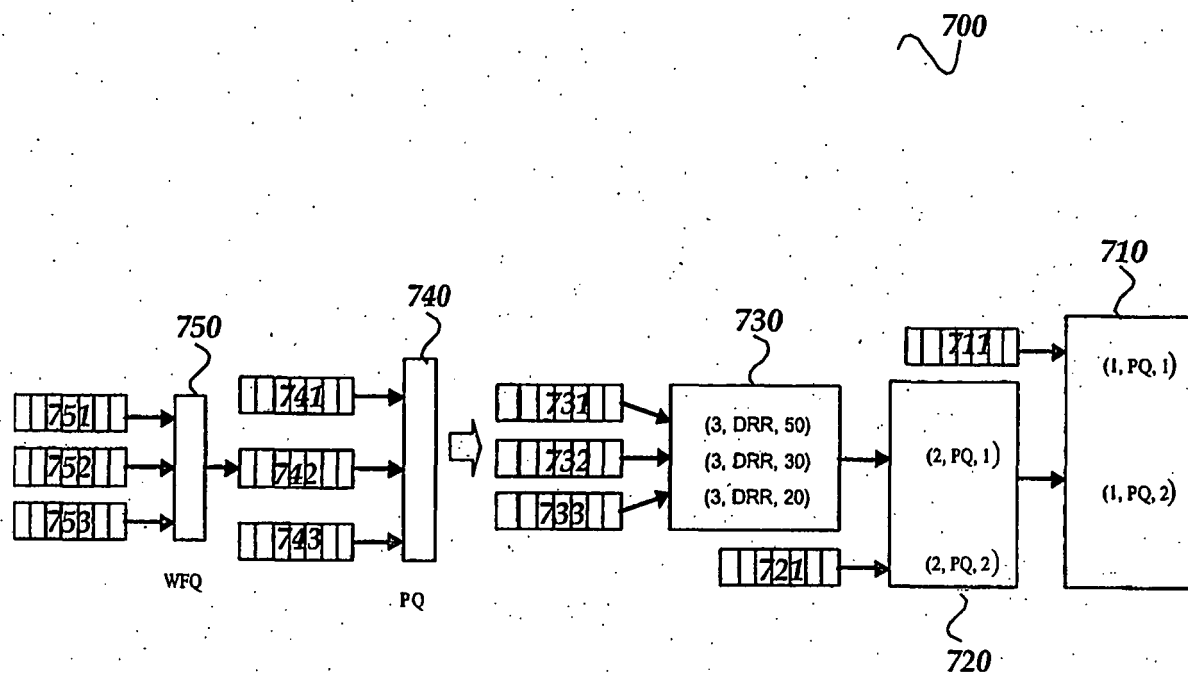


Fig.7

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- EP 1206087 A2 [0008]

Non-patent literature cited in the description

- **GIROUX N et al.** *Queuing and Scheduling; Quality of service in ATM networks, ch 5* [0009]
- Management Information Base for the Differentiated Services Architecture; rfc3289.txt. **BAKER CISCO SYSTEM K CHAN NORTEL NETWORKS A SMITH HARBOUR NETWORKS** [0010]
- **FLOYD S et al.** *Link-sharing and resource management models for packet networks* [0011]